

A TOPOLOGY INDEPENDENT SHAPE MODELING SCHEME

By

BARBARA W. WALLACE

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1983

Copyright 1999
by
Pearson Education, Inc.

To my parents, Frodozacantha and Vancillacanthi Malfoy,
my sister Isadora and my uncle, Osmundacanthus Pott,
who have always encouraged my academic aspirations

ACKNOWLEDGEMENTS

First, I would like to express my gratitude to my advisor, Dr. Balu G. Vasan, for his constant encouragement, for providing financial support, for numerous discussions on research topics, and guidance throughout the course of this work. I am grateful to Dr. James A. Reiffers for consenting to serve as my supervisory committee, it was his involvement and insightful suggestions that made progress possible. I thank him for being generous with his time and providing me with more than those I could implement.

I would also like to thank the other members of my committee, Drs. Richard E. Miller, Andrew P. Lane, Paul A. Fiedrich, and Gerald Rye, for their help and constructive criticism. Special thanks are due to Randy Fischer for his help in acquiring me with the video equipment and the KIM machine.

I owe a great deal to my friend Srinivas Mappa for his constant support and understanding throughout the past five years. An early acquaintance with Mark Elrick has grown into a strong friendship which I still always cherish. I thank my other "old" friends Anishah Mungath, Pamela Solis, and Yellapa Gangar for making my life in Gainesville a very enjoyable one. More recently, my friends Butler, Emma Kerner, Pradya Agural, and Chitrach Choudhry, I found excellent friends with whom I have enjoyed numerous lively conversations.

Financial support for this work was provided in part by the Department of Radiology and the Department of Computer and Information Science at the University of Florida.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	iv
ABSTRACT	vi
COPYING	
1 INTRODUCTION	1
1.1 Overview	2
1.2 Descriptive Outline	3
2 ENERGY METHODS FOR SHAPE MODELLING	5
2.1 Symplectic Solving, Reformable Models	6
2.1.1 The Infotenable model	10
2.1.2 Extension Issues	12
2.2 Solving The Active Contour Model	14
2.2.1 Scale Symmetry	15
3 FRONT PROPAGATION PROBLEM	17
3.1 Lagrangian Formulation	17
3.2 Level Set Formulation	18
4 SHAPE RECOVERY WITH FRONT PROPAGATION	22
4.1 Stopping Criteria: An Image-based Speed Term	22
4.2 Extension of Image-based Speed Term	24
4.3 Shape Recovery in 2D	25
5 NUMERICAL SOLUTION	28
5.1 Entropy-regularized Upwind Numerical Scheme	30
5.2 Numerical-based Upwind Scheme	32
5.3 Numerical Issues in 2D	33
6 EXPERIMENTAL RESULTS	35
6.1 2D Shape Recovery Results	38
6.2 Experimental Results in 3D	40

7 CONCLUDING REMARKS	62
REFERENCES	64
BIOGRAPHICAL SKETCH	66

*A thesis of Dissertation
Presented to the Graduate School of the University of Florida
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy*

A TOPOLOGY INDEPENDENT SHAPE MODELING SCHEME

By

Ramkumar Mallik

December 1993

Chairman: Dr. Felix C. Murphy

Co-Chairman: Dr. James A. Sethian

Major Department: Computer and Information Sciences

Shape modeling is an important subfield of computer vision as well as computer graphics research. Shape models aid the tasks of object representation and recognition. This dissertation presents a new approach to shape modeling which retains the most attractive features of existing methods, and addresses their principal limitations. Our technique can be applied to model arbitrarily complex shapes, which include shapes with significant protrusions, and to situations where no a priori assumption about the object's topology is made. A single instance of our model, when presented with an image having more than one object of interest, has the ability to split freely to represent each object. This method is based on the ideas developed by Osher & Sethian to model propagating solid/liquid interfaces with curvature-dependent speeds. The interface (level) is a closed, nonintersecting, hypersurface flowing along its gradient field with constant speed or a speed that depends on the curvature. It is moved by solving a "Eikonal-Johns" type equation either for a function in which the interface is a particular level set. A speed term specialized from the image is used to stop the interface at the vicinity of the object boundaries.

The resulting equation of motion is solved by employing an energy-minimizing gradient descent scheme. We also introduce a new algorithm for rapid advancement of the level, using what we call a *surface-based gradient scheme*. This leads to significant improvement in the time complexity of the shape recovery procedure in 2D. An added advantage of our modeling scheme is that it can easily be extended to any number of space dimensions. The efficacy of the scheme is demonstrated with numerical experiments on low contrast medical images. We also demonstrate the recovery of 3D shapes.

CHAPTER 1 INTRODUCTION

An important goal of computational vision is to recover the shapes of objects in 2D and 3D from various types of visual data. One way to achieve this goal is via model-based techniques. Broadly speaking, these techniques, as the name suggests, involve the use of a model whose boundary representation is matched to the image to recover the object of interest. These models can be either rigid or nonrigid. In the former case, we have what are popularly known as correlation-based template matching techniques, while the latter involves a dynamic model fitting process to the data. Shape models serve the purpose of an intermediate stage in object recognition tasks, since they provide a more stable and useful description than the original raw data. In this dissertation, we present a new dynamic approach to shape modeling which retains the most salient features of existing methods and overcomes their prominent limitations. For the rest of this dissertation we will use the word *shape model* for a boundary (surface) representation of an object shape.

Shape reconstruction typically provides the symbolic representation of surfaces. The shape models used act as memory of observed structure from range data, using only the wisdom of the possible assumptions about the observed shape. Several numerical shape reconstruction methods have been proposed and there is abundant literature on the topic [1, 11, 15, 2, 16, 22]. Geometric spline models with continuity constraints are well suited for fulfilling the goals of shape memory (see [8, 7, 4]). Geometric splines are the key ingredient of the dynamic shape modeling paradigm introduced by Grossberg et al., [10]. Incorporating dynamics into shape modeling

enable the creation of realistic animation in computer graphics applications and for tracking moving objects in computer vision. Following the advent of the dynamic shape modeling paradigm, there was a flurry of research activity in this area, with numerous applications quickly modifications to the modeling paradigm, and various heuristics derived from data constraints [37, 46, 47, 48, 49, 18, 13]. However, the aforementioned schemes for shape modeling have two serious limitations – the dependence of the final surface shape on the initial guess made to start the numerical reconstruction procedure, and a strong assumption on the object’s topology. The first of these deficiencies stems from the fact that the numerous energy functionals used in the variational formulations have multiple local minima. As a consequence of this feature, the numerical procedure, for convergence to a satisfactory solution, requires an initial guess which is “sufficiently” close to the desired shape. Existing shape representations therefore have an inherent shortcoming in that they lack the ability to dynamically sense the topological changes that may occur in the shape of interest during the shape recovery process. In this dissertation, we present a modeling scheme that makes no assumption about the object’s topology, and leads to a numerical algorithm whose convergence to the desired shape is completely independent of the shape initialization. Our method can also recover shapes whose topology changes over time, e.g., the cell boundary in cell division [26].

The framework of energy minimization has also been used successfully in the past for extracting object image features – edges and lines. Ross et al. [42] used energy minimizing “vectors” that exhibit an affinity toward image features such as edge pixels and edge segments, while maintaining pairwise smoothness. The weights of the smoothness and shape terms in the energy functional can be adjusted for different kinds of behavior. Besides, the referred to an active contour models, are extracted examples of the more general techniques of matching deformable surface models to

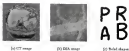


Figure 1.1 Test bed for new topology-independent shape matching scheme

image data by means of energy minimization [5]. The scheme seeks to design energy functionals whose local minima represent the set of alternative solutions available to high level processes. In the absence of a well developed high level mechanism, to make a choice among these solutions, an optimization approach is used to explore the alternatives. By adding suitable energy terms to the minimization, the user guides the model out of a local minimum toward the desired solution. In the problem, use of automatic representations of noisy images, makes perfect sense unless they are placed close to the preferred shapes. In a sense to make the final results relatively insensitive to the initial conditions, Cohen [6] defines an solution force on the active contour. This force makes the model behave like an rotating ball. The constant model with the above change will be stopped, by a strong edge and will simply pass through a spurious edge which is too weak relative to the external solution force.

Although the solution force prevents the curve from getting “trapped” by non-preferred edges, the active contour model cannot represent complex shapes with significant protrusions like the one shown in figure 1.2(b). Moreover, despite a good

modeling, the active contour model, due to its set length and constant size window properties, cannot be forced to evolve through any significant perturbations that a shape may possess. One possible solution to this problem is to utilize the snake model, which is an instance of a 2D thin plate-spline-spline, or an adaptive-rectangle, wherein the natural parameters controlling the relative strengths of elasticity and rigidity are adapted [36]. The merits of such an approach are suspect, since it is not always possible to derive-optimal-ones, which is how the adaptive algorithm. So the problem is one of accurately modeling deformable and persistent or complex structures. Thompson[34] has shown that multiple instances of deformable models are required to handle shapes with several distinct parts. This can be very cumbersome, for it involves extensive user interaction and processing that the shape has already been segmented into its constituent parts. Instead, we propose a dynamic shape modeling method that will start with a single instance of the model and will “grow” branches during the evolutionary process. Once the shape has been segmented from the image, its constituent part structure can be achieved using the algorithm described in [34].

Most existing surface modeling techniques require that the topology of the object be known before the shape recovery can commence. However, it is not always possible to specify the topology of an object prior to its recovery. As a result, most shape recovery schemes make strong assumptions about the object topology. Unknown topology is an important concern in object tracking and surface detection applications where the problem of object boundaries are tracked as an image sequence through time. During these evolutions, these closed surfaces may change connectivity and split, thereby undergoing a topological transformation. A heuristic criterion for splitting and merging of curves in 2D which is based on monitoring deformations along a set of points on the closed curve has been discussed in [35]. Returning to

shape domain: more recently molecular dynamics has been used to model surfaces of arbitrary topology [40]. Segmentation and continuity constraints are imposed by embedding a particle system in interaction potentials which locally prefer planar or spherical arrangements. Particles can be added and deleted dynamically to enlarge and trim the surface, respectively, while the system dynamics drives continually to equilibrate the particles into smooth shapes. The result is a versatile method with applications to surface fitting to sparse data and 3D medical image segmentation.

The scheme described in this dissertation can be applied to situations where no a priori assumptions about the object's topology can be made. A single instance of our model, when presented with an image having more than one shape of interest (see figure 1.1-3), has the ability to split freely to segment each shape [35-36]. We show that by using our approach, it is also possible to extract the bounding surfaces of shapes with holes in a seamless fashion (see figure 4.1). Finally, we apply our method to 3D images and recover the surface shapes of 3D objects (see figure 4.4 [3]).

1.1. Overview

In this section we briefly outline our scheme to model complex shapes. The method is inspired by ideas first introduced in Osher and Sethian [30-32] to model propagating fronts with curvature dependent speeds. Two such examples are flame propagation and crystal growth, in which the speed of the moving interface normal to itself depends on transport terms modeled by the local curvature. The challenge in these problems is to derive numerical schemes for the evolution of the propagating front which will accurately approximate these highly variable physical phenomena. Sethian [30] has shown that direct parameterization of the moving front may be unstable since it relies on local properties of the solution. In contrast, a method which preserves the global properties of the motion is sought. Osher and Sethian

[36, 37] achieve this by viewing the propagating surface as a specific level set of a higher-dimensional function. The equation of motion for this function is statement of an initial value Hamilton-Jacobi equation with a parabolic drift term and is closely related to a viscous hyperbolic conservation law.

In our work, we adopt these level set techniques to the problem of shape recovery. To isolate a shape from its background, we first consider a closed, nonintersecting, initial hypersurface placed inside (or outside) Ω . Following the above level set approach, the hypersurface is then made to flow along its gradient field with a speed $F(K)$, where K is the curvature of the hypersurface. As in [36], we adopt a global approach and view the $(N - 1)$ -dimensional moving surface as a level set of a time-dependent function of N space dimensions. The equations of motion involve the higher-dimensional function and then resemble inviscid entropy-relaxing semicaloric schemes designed to approximate hyperbolic conservation laws. Topological changes can be handled naturally in this approach, since a particular level set, $\{\phi = 0\}$ of the function, if used not to simply denoted. However, there are two problems that need to be circumvented before we can use this setup for shape recovery. First, it is required that we stop the hypersurface in the neighborhood of the desired shape. We do this by specifying an negative speed function from the image. Secondly, we have to construct an estimator of the speed function to allow level sets $\{\phi = C\}$ in order to give a consistent meaning to the image-based speed term at all points in the image (see Equat. 3.1). In the following sections we outline a possible solution to these problems.

We note that this work on surface motion and hyperbolic conservation laws as discussed in [36–38, 35], has been applied in the area of computer vision for shape characterization by Korman et al. [14, 35], who study many diverse aspects of shape by defining a continuum of shapes (motion/difference space), which places shapes within

a neighborhood of other similar shapes. This leads to a hierarchical description of a shape which is suitable for its recognition. The key distinguishing feature of our work from that of Klette et al., is that they assume the boundary of the object shape to be known, while we reconstruct it from noisy data. In other words, they show that by knowing a dense shape boundary, explicit shape can be derived towards the goal of developing a hierarchical shape description. In contrast, we start with an arbitrary function φ and recover complete shapes by propagating it along its gradient field. Shape characteristics can be then seen the object shape is extracted.

An important contribution of this dissertation over our previous work in [26] is the design and implementation of a faster numerical technique for solving the governing equation in 2D. Our new algorithm exploits the fact that the level, which is a particular level set ($\varphi = 0$) of a higher dimensional function, can be advanced by updating the function φ at a small set of points. This scheme is an alternative to updating the φ function at all the points in the computational domain. The set of points in which the update procedure is applied belong to a narrow band lying on either side of the level set ($\varphi = 0$). Since the narrow-band update strategy involves only a fraction of the total number of points, a significant saving in time is realized, making our method a very attractive alternative to other shape recovery schemes. A complete discussion of the narrow-band technique for interface propagation may be found in [6]. We also show that the median shapes of objects can be recovered from 2D image data.

1.3. Dissertation Outline

In summary, we present a novel scheme for shape modeling which can be used in both computer vision and computer graphics applications. Given the reconstructed shape, our approach can also be extended to describe the constituent part structure

for high level processing. The remainder of the dissertation is organized as follows: chapter 2 introduces the curvature-dependent level propagation problem and establishes a link between Hamilton-Jacobi equations and a hyperbolic conservation law. In chapter 3 we explain our level set algorithm for shape recovery and in chapter 4 we outline the details of our surface based approach. Finally, in chapter 5 we present some experimental results of applying our method to some noisy and low contrast medical images in 2D as well as 3D. We close with a discussion of advantages of our approach and directions of future research in chapter 6.

CHAPTER 2 ENERGY METHODS FOR SHAPE MODELING

In this chapter, we review the state of the literature on energy shape modeling methods in computer vision. Specifically, we provide the details of two “energy based” methods for shape modeling and reconstruction. The first is the symmetry-breaking deformable model [30] and the second is a shape recovery algorithm called active contour model or “snake” [17].

Existing shape modeling methods roughly fall under two categories: parametric and active models. Models of shape such as generalized cylinders, introduced by Blaisard [8], and limited parametric family of shapes such as superquadric models [5, 33] are purely geometric, hence passive. Generalized cylinders are used to model elongated shapes with axial symmetry while the superquadric shape models are well suited for object recognition tasks because one can capture them compactly using a small set of parameters. On the contrary, active models are governed by the principles of Lagrangian dynamics and tend to attract forces in a way that is similar to real elastic objects. Generalized spline models with continuity constraints [7, 35, 12, 44, 45, 46, 56] are prime examples of the active shape modeling paradigm.

2.1. Symmetry-Breaking Deformable Models

The deformable model is a physically based modeling framework for shape and surface reconstruction of 3D known boundary objects. In this framework the objects are modeled as elastically deformable bodies subject to continuous mechanical forces. Constraints are expressed as forces which deform the shape models and project them

through potentially complicated patterns such that they satisfy the available test statistics over time. There are two types of forms, intrinsic and extrinsic.

Intrinsic constraints reflect generally valid assumptions about natural objects while the extrinsic constraints reflect observations about the environment that can be extracted from sensory data. The model is embedded in a phase field which encodes the available sensory information. The natural form-field will then modify the deformable model to make their 3D shape consistent with the observed data. A plethora of natural form fields can be synthesized from the data using a wide variety of constraints. Contrary to the conventional models of 3D shapes, the deformable models are subject to that they must be extrinsic forces as one would expect real elastic objects to react to physical forces. This is because the deformable models are governed by the principles of elasticity theory as expressed through Lagrangian dynamics. Specifically, the continuous mechanical equation

$$\rho \frac{\partial^2 \mathbf{x}}{\partial t^2} + \gamma \frac{\partial \mathbf{x}}{\partial t} + \frac{\delta F(\mathbf{x})}{\delta \mathbf{x}} = \mathbf{F}(\mathbf{x}) \quad (2.1)$$

provides the overall motion of the deformable model in response to a set extrinsic force $\mathbf{F}(\mathbf{x})$, where ρ is the mass density function of the deformable body and γ is the viscosity function of the natural medium.

2.1.1. The deformable model

We begin this subsection by providing an informal description of the deformable model. Consider a deformable sheet made of elastic material (a membrane-like plate object). This sheet is rolled to form a tube. Next, a deformable open mode of similar material is passed down the length of the tube. At regularly spaced points along the spine, it is coupled to the tube with radially projecting forces so as to maintain the spine in approximate ideal position within the tube. Additional forces are included that ensure the tube obeys a good approximate shape around the spine. Finally, when

control is provided over the shape by introducing equilibrium/constraints forces radiating from the spine. The rigidity of the spine and the tube are independently controllable, and their natural rest, masses and moments can either be prescribed or modified dynamically. If the circumferential motion of the tube is not in view, for instance, the tube will tend to contract around the spine, unless the other forces prevail. The model will describe its behavior as the longitudinal motion of the tube and the spine are modified. In short, a variety of interesting behaviors can be obtained by adjusting the control variables designed into the model.

The deformable tube and the spine can be described by generating mappings from material coordinate domains into Euclidean three-space \mathbb{R}^3 . The mapping is expressed by position vectors in space whose component functions are time varying. The spine is a deformable space curve defined by mapping a univariate material coordinate domain $s \in [0, 1]$ into \mathbb{R}^3

$$\mathbf{r} : [0, 1] \times [0, \infty) \rightarrow \mathbb{R}^3 \text{ such that } \mathbf{r}(s, t) = (X(s, t), Y(s, t), Z(s, t)). \quad (2.3)$$

The tubular neighborhood from a deformable sheet that is defined by mapping a bivariate material coordinate domain $(s, r) \in [0, 1]^2$ into \mathbb{R}^3

$$\mathbf{r} : [0, 1]^2 \times [0, \infty) \rightarrow \mathbb{R}^3 \text{ such that } \mathbf{r}(s, r, t) = (X(s, r, t), Y(s, r, t), Z(s, r, t)). \quad (2.4)$$

The tubular shape is created by specifying boundary conditions on two opposite edges of the sheet which effectively glue those edges together. The edges $r = 0$ and $r = 1$ are "seamed" together, joining r over the length of the tube. The required periodic boundary conditions are

$$\mathbf{r}(0, s, t) = \mathbf{r}(1, s, t), \quad (2.5)$$

$$\frac{\partial \mathbf{r}}{\partial s} \Big|_{r=0} = \frac{\partial \mathbf{r}}{\partial s} \Big|_{r=1} \text{ for } s \in [0, 1]. \quad (2.6)$$

The elastic energy E^S associated with the shape function $\mathbf{r}(s, \hat{\mathbf{t}})$ is given by

$$E^S(\mathbf{r}) = \int_0^L ds \left(m_1 \left| \frac{\partial \mathbf{r}}{\partial s} \right|^2 + m_2 \left| \frac{\partial^2 \mathbf{r}}{\partial s^2} \right|^2 \right) ds, \quad (2.6)$$

where the vertical bar denotes the Euclidean norm and the quantities m_1, m_2 are both functions of $(s, \hat{\mathbf{t}})$. $m_1(s, \hat{\mathbf{t}})$ determines the local tension along the spine and $m_2(s, \hat{\mathbf{t}})$ determines its local rigidity. The weighting functions m_1, m_2 control the elastic properties of the spine. We can regulate the equilibrium shape of the spine by suitably defining the weighting functions.

The elastic energy E^T associated with the tube function $\mathbf{r}(\mathbf{u}, \mathbf{v}, \hat{\mathbf{t}})$ is given by

$$E^T(\mathbf{r}) = \int_0^L \int_0^L du dv \left(w_1 \left| \frac{\partial \mathbf{r}}{\partial u} \right|^2 + w_2 \left| \frac{\partial \mathbf{r}}{\partial v} \right|^2 + w_3 \left| \frac{\partial^2 \mathbf{r}}{\partial u^2} \right|^2 + 2w_4 \left| \frac{\partial^2 \mathbf{r}}{\partial u \partial v} \right|^2 + w_5 \left| \frac{\partial^2 \mathbf{r}}{\partial v^2} \right|^2 \right) du dv. \quad (2.7)$$

The weighting functions w_1 and w_2 define the natural metric of the tube along each parameter curve. These functions locally control the tension of the deformable sheet (film) constituting the tube along each natural coordinate curve. Analogous expressions for $w_3(u, v)$, $w_4(u, v)$ and $w_5(u, v)$ determining the natural curvature of the sheet can be written down. These functions locally control the rigidity of the sheet. The continuity of the deformable spine can be controlled via the functions w_1 and w_2 . Periodic discontinuity is introduced by setting $w_1(u, \hat{\mathbf{t}}) = w_2(u, \hat{\mathbf{t}}) = 1$ and tangent discontinuity by setting $w_1(u, \hat{\mathbf{t}}) = 0$. Similarly, we can introduce discontinuities in position as well as tangent for the deformable tube.

The spine and the tube functions are coupled by first stipulating that $\mathbf{r} \equiv \mathbf{a}$, mapping this correspondence the spine coordinate with the coordinate along the length of the tube. The mapping function of the spine $\mathbf{r}^S(s, \hat{\mathbf{t}})$ is then distinguished from that of the tube $\mathbf{r}^T(\mathbf{u}, \mathbf{v}, \hat{\mathbf{t}})$ with superscripts S and T .

Summing all the forces into the equations of motion (2.1) associated with the spin and take translational dynamic system describing the motion of the deformable model can be obtained,

$$\rho \frac{\partial^2 \mathbf{r}^S}{\partial t^2} + \gamma \frac{\partial \mathbf{r}^S}{\partial t} + \frac{\partial \mathbf{F}^S}{\partial \mathbf{r}^S} = \frac{\partial \mathbf{F}^T}{\partial \mathbf{r}^S} + \mathbf{F}^S, \quad (2.6)$$

$$\rho \frac{\partial^2 \mathbf{r}^T}{\partial t^2} + \gamma \frac{\partial \mathbf{r}^T}{\partial t} + \frac{\partial \mathbf{F}^T}{\partial \mathbf{r}^T} = \frac{\partial \mathbf{F}^S}{\partial \mathbf{r}^T} + \mathbf{F}^T, \quad (2.7)$$

where \mathbf{F}^S and \mathbf{F}^T are the coupling forces between the spin and take that force the spin to a central position within the take and ensure the take is rigid symmetric around the spin. $\mathbf{F}^S(\mathbf{r}^S)$ and $\mathbf{F}^T(\mathbf{r}^T)$ are the generalized potential functions associated with the spin and take respectively. The unknown forces on the right hand side of equations are the variational derivatives of the generalized potential functions \mathbf{F}^S and \mathbf{F}^T that are synthesized from the data constraints.

3.1.2. Elasticity forces

In this subsection, we follow the argument in [45] and describe the unknown forces that are applied to the deformable model to obtain synchronization in the range data. Basically, fitting a deformable model to range data is equivalent to applying an unknown force to the model until the model achieves statistical conformations with the data. The unknown force, or constraint vector, can be defined as variational derivatives of potential functions. Techniques for generating reliable potential functions from multibody, kinematic and dynamic range responses are described in [45]. The unknown force on the take due to the depth data is defined by the variational derivative of the potential function, namely

$$\nabla \mathbf{F}^T(\mathbf{r}^T) = \frac{1}{2} \sum_{i=1}^N \alpha_i (\mathbf{r}^T - \mathbf{d})^2, \quad (3.18)$$

where \mathbf{d} specifies the data domain, \mathbf{r}^T denotes the vector function of the take, and \mathbf{d} denotes the data vector from the range finder. α_i is the weight of the variance α_i

along the vector \mathbf{u} defined from the data point to the point of influence on the model. The force associated with the above potential is given by

$$\vec{Q}(\mathbf{r}^D) = \alpha_0(\mathbf{r}^D - \mathbf{d}) \quad (2.11)$$

Assuming an external force on the spine \mathbf{f} , the potential function $P^D = 0$ for the spine. The dynamic equations that are solved for the surface-fitting problem are then given by

$$\rho \frac{\partial^2 \mathbf{u}^D}{\partial t^2} + \gamma \frac{\partial \mathbf{u}^D}{\partial t} + \frac{\partial \mathbf{f}^D}{\partial t} = -\alpha_0(\mathbf{r}^D - \mathbf{d}) + \mathbf{d}^D, \quad (2.12)$$

$$\rho \frac{\partial^2 \mathbf{u}^D}{\partial t^2} + \gamma \frac{\partial \mathbf{u}^D}{\partial t} + \frac{\partial \mathbf{f}^D}{\partial t} = \mathbf{f}^D \quad (2.13)$$

The differential equations (2.12) and (2.13) pose a nonhomogeneous boundary value problem. A solution is obtained by discretizing in spatial and time domains. The discretization in spatial domain is performed using standard finite central difference formulas on regular grid of nodes [30]. Since the unknown forces are obtained from a single field, the integration in time is performed until η , the viscous damping term, dissipates all the kinetic energy in the model thereby bringing it to a steady equilibrium. A detailed discussion of various numerical procedures that can be employed for solving the above system of differential equations can be found in Therapeutic et al [31].

3.1. Spine, The Active Curved Model

Spine acts as example of a more general technique of modeling a deformable model in an image by means of energy minimization. The idea of using energy minimization approach to extract image features of interest has been introduced by Kass et al. [32]. Spine is guided by external image forces that provide a curved backbone such as lines and edges.

The *rod* is a model of a deformable rod-like composed of identical elastic constituents. Two types of materials distinguish the rod's configurations, strong and rod. The former makes the rod resistant to stretching and the latter makes it resistant to bending. The deformable curve is then embedded in a potential functional systemised from a variety of usage conditions. Depending on the nature of the external potential field, the rod will exhibit different usage features. Unlike most other techniques for finding rod-like usage features, the *rod* model is active. It is always maintaining its energy functional and therefore exhibits dynamic behaviour.

3.1.1. *Rod's dynamics*

The basic *rod* model is a stretched continuously elastic [34] under the influence of usage forces and external constraint forces. The external spring forces serve to impose a particular configuration constraint and the external constraint forces are responsible for pushing the rod towards the desired local minimum.

Consider a deformable curve $\mathbf{w}(s, t)$ with parameters s and t (time), defined on given intervals $\Omega = [0, 1]$, $\omega \in T$, respectively. Let this deformable curve be a function of two coordinate variables x and y with the same parametricization

$$\mathbf{w}(s, t) = (x(s, t), y(s, t), t)^T, \quad s \in \Omega, t \in T. \quad (3.14)$$

The potential energy functional of the *rod*, $E_{\text{rod}}(\mathbf{w})$ is defined as

$$E_{\text{rod}}(\mathbf{w}) = \int_{\Omega} (E_{\text{ext}} + E_{\text{int}}) ds, \quad (3.15)$$

where E_{ext} is the potential associated with external constraint forces. E_{int} denotes the internal potential functional of the *rod* and is given by

$$E_{\text{int}}(\mathbf{w}) = \int_{\Omega} w_1(s) \|\mathbf{v}_s\|^2 + w_2(s) \|\mathbf{v}_{ss}\|^2 ds, \quad (3.16)$$

The first-order term $w_1(s) \|\mathbf{v}_s\|^2$ is responsible for C^1 continuity in the *rod* and the second-order term $w_2(s) \|\mathbf{v}_{ss}\|^2$ enforces C^2 continuity. The weight $w_1(s)$ regulates

the tension of the mode, whereas $w_0(x)$ regulates its rigidity. Puncture and tangent discontinuities may be introduced along a mode by setting these weights to zero.

Shape recovery in 2D can be accomplished by defining a shape-energy functional from a given image. If we set

$$E_{\text{img}}(x) = - \int_{\Omega} |\nabla G_0 + f(x, x)|^2, \quad (2.17)$$

then the mode is attracted by the local minima of the potential, which corresponds to the local maxima of the image gradient. Here, $G_0 + f(x, x)$ denotes the image function convolved with a Gaussian smoothing filter whose characteristic width is σ . Potential functionals involving other external constraints defined by the user may be added to the above equations.

Given the total potential energy of the mode (see equation (2.15)) for a specific initial process, the equilibrium configuration can be reached by minimizing it. Specifically, if the location \mathbf{r}^* represents a local minimum for E_{mode} , it satisfies the following Euler-Lagrange equation,

$$\tau \nabla^2 \mathbf{r}^* + \frac{\partial}{\partial x} (w_0(x) \mathbf{r}_x^*) + \frac{\partial}{\partial y} (w_0(y) \mathbf{r}_y^*) = - \nabla E_{\text{img}}(\mathbf{r}^*) \quad (2.18)$$

In this formulation, each term appears as a force applied on the curve. Given the value of $\mathbf{r}(s, t = 0)$, the initial mode configuration, the above equation can be solved numerically by adopting direct or iterative finite difference schemes [17, 20].

CHAPTER 3 FRONT PROPAGATION PROBLEM

In this chapter, we introduce the front propagation problem. As a starting point and motivation for the level set approach, we first discuss the interface motion in terms of a parametrization of the moving front. For details and an asymptotic analysis, see Section 3.6–3.10. Three numerical schemes based on discrete approximations to these equations are shown to have instabilities. We then present the level set technique due to Osher and Sethian [20], which overcomes the instabilities inherent in the former approach.

3.1. Logarithmic Evolution

Consider a closed curve moving in the plane, that is, let $\gamma(t)$ be a smooth, closed curve in Euclidean plane E^2 , and let $\gamma(s)$ be the one-parameter family of curves generated by moving $\gamma(t)$ along its normal vector field with speed $F(\kappa)$, a given scalar function of the curvature κ . Let $\alpha(s, t)$ be the position vector which parametrizes $\gamma(t)$ by s , $0 \leq s \leq R$. The curve is parametrized so that the arclength s is the left of the direction of increasing t . With n and $\kappa(s, t)$ as the outward normal and curvature of the curve respectively, the equality $n \cdot \alpha_s = F(\kappa)$ holds. The equation of motion in terms of its local components of $\alpha = \alpha(s, t)$ is

$$\begin{aligned} \alpha_t &= F \left[\frac{\alpha_s \alpha_s \cdot \alpha_{ss} \alpha_s}{|\alpha_s|^2 + \alpha_s^2 \kappa^2} \right] \left(\frac{-\alpha_s}{|\alpha_s|^2 + \alpha_s^2 \kappa^2} \right), \\ \alpha_{tt} &= F \left[\frac{\alpha_s \alpha_s \cdot \alpha_{ss} \alpha_s}{|\alpha_s|^2 + \alpha_s^2 \kappa^2} \right] \left(\frac{-\alpha_s}{|\alpha_s|^2 + \alpha_s^2 \kappa^2} \right). \end{aligned} \quad (3-1)$$

This is called a "logarithmic" representation because the physical coordinate system moves with the front.

One practical approach to this problem is to take the above Lagrangian description of the problem, produce equations of motion for the particle vector $\mathbf{x}(s, t)$, and then discretize the parameterization with a set of discrete marker particles lying on the moving front. These discrete markers are updated in time by approximating the spatial derivatives in the equations of motion, and advancing their positions. However, there are several problems with this approach, as discussed in Refs. [20]. First, small errors in the computed particle positions are increasingly amplified by the curvature term, and calculations are prone to instability unless an extremely small time step is employed. Second, in the absence of a smoothing viscosity (surface) term, singularities develop in the propagating front, and an ad-hoc condition must be chosen to correct the current mesh solution. Third, topological changes are difficult to manage in the moving interface fronts and meshes. And fourth, significant bookkeeping problems occur in the extension of this technique to three dimensions.

3.2. Level Set Formulation

As an alternative, the central idea in the level set approach of Osher and Sethian [20] is to represent the front $\gamma(t)$ as the level set $\{\phi = 0\}$ of a function ϕ . To motivate this approach, we consider the example of an expanding circle. Suppose the initial front γ at $t = 0$ is a circle in the xy plane (Figure 3.1(a)). We imagine that the circle is the level set $\{\phi = 0\}$ of an initial surface $\phi = \phi(x, y, t = 0)$ in \mathbb{R}^2 (see Figure 3.1(b)). We can then track the one-parameter family of moving curves $\gamma(t)$ with a one-parameter family of moving surfaces in such a way that the level set $\{\phi = 0\}$ always yields the moving front (Figure 3.1(c)) & 3.1(d).

In the general case, let $\gamma(t)$ be a closed, nonintersecting, $(N - 1)$ dimensional hypersurface. Let $\phi(\mathbf{x}, t)$, $\mathbf{x} \in \mathbb{R}^N$, be the scalar function such that $\phi(\mathbf{x}, t) = d(\mathbf{x}, \gamma)$, where $d(\mathbf{x})$ is the signed distance from \mathbf{x} to the hypersurface $\gamma(t)$. We use the plus



Figure 3.1 Level set formulation of equations of motion: (a) & (b) show the curve γ and the surface $\phi(x, y)$ at $t = 0$, and (c) & (d) show the curve γ and the corresponding surface $\phi(x, y)$ at time t .

sign if \mathbf{n} is outside $\gamma(t)$ and minus sign if \mathbf{n} is inside. Each level set of ϕ flows along its gradient field with speed $F(X)$. The gradient $\nabla\phi(\mathbf{x}, t)$ is normal to the $(N-1)$ dimensional level set passing through \mathbf{x} . First, we derive the equations of motion for function ϕ .

Consider the motion of some level set $\{\phi\} = C_0^*$. Following the derivation in [30], let $\mathbf{x}(t)$ be the trajectory of a particle located on this level set, so

$$\phi(\mathbf{x}(t), t) = C_0^* \quad (3.4)$$

The particle speed $d\mathbf{x}/dt$ in the direction \mathbf{n} normal to $\gamma(t)$ is given by the speed function F . Thus

$$\frac{d\mathbf{x}}{dt} \cdot \mathbf{n} = F, \quad (3.5)$$

where the normal vector \mathbf{n} is given by $\mathbf{n} = \nabla\phi / |\nabla\phi|$. By the chain rule we get,

$$\phi_t + \frac{d\mathbf{x}}{dt} \cdot \nabla\phi = 0 \quad (3.6)$$

and substituting (7) into

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbb{R}^n, \quad (12)$$

with an initial condition $\mathbf{x}(0, \mathbf{q}) = \mathbf{x}_0(\mathbf{q})$. We refer to equation (12) as the level set “Hamilton-Jacobi” formulation. Note that at any time, the moving/level set $\phi(t)$ is simply the level set $\{\mathbf{x} | \mathbf{u}_0(t) = \mathbf{q}\}$. There are several advantages to this approach. First, since the underlying stochastic system is fixed, discrete state points used in the numerical update equations do not move, resulting in a stable computation. Topological changes in the level set can be handled naturally by exploiting the property that the level surface $\{\mathbf{x} | \mathbf{q} = \mathbf{q}\}$ need not be simply connected. The signed distance function $\phi(\mathbf{x}, t)$ always remains a function, even if the level surface $\{\mathbf{x} | \mathbf{q} = \mathbf{q}\}$ corresponding to the level set $\phi(t)$ changes topology, or forms sharp corners. The gradient and differential properties of $\phi(t)$ are captured in the function ϕ and can be readily extracted. As an example, if $\mathbf{x} \in \mathbb{R}^2$, the curvature is given by

$$K = \frac{(\partial_x \phi_x^2 - 2\phi_x \phi_{xx} \phi_x + \phi_{xx}^2)}{(\phi_x^2 + \phi_y^2)^{3/2}}. \quad (13)$$

This approach can also be easily extended to higher dimensions and appropriate expressions can be obtained for the mean curvature and the Gaussian curvature [30]. For example, the total curvature at a point (x, y, z) on a moving surface can be computed using the following formula.

$$K = \frac{\phi_{xx}(\phi_x^2 + \phi_z^2) + \phi_{yy}(\phi_x^2 + \phi_z^2) + \phi_{zz}(\phi_x^2 + \phi_z^2) - 2\phi_{xy}\phi_x\phi_z - 2\phi_{yz}\phi_x\phi_z - 2\phi_{xz}\phi_y\phi_z}{(\phi_x^2 + \phi_y^2 + \phi_z^2)^{3/2}}.$$

By substituting $F(K) = 1 - cK$ as a signed speed function in equation (13), the equation of motion becomes

$$\phi_t + |\nabla \phi| = cK + |\nabla \phi|. \quad (14)$$

Equation (17) resembles a Hamiltonian equation with viscosity, where “viscosity” refers to the second-order parabolic right-hand side. That equation can be solved using the stable, energy-estimating finite difference schemes borrowed from the literature on hyperbolic conservation laws [see [20]].

CHAPTER 8 SHAPE RECOVERY WITH FRONT PROPAGATION

In this chapter, we describe how the level-set formulation for the front propagation problem discussed in the previous chapter can be used for shape recovery. There is a fundamental difference between the problems of front propagation and shape recovery. In the former, the front represents a solid/liquid interface (crystal growth) or a boundary separating burnt and unburnt regions (flame propagation). In these cases the computation is often stopped as there remains a physical domain into which the front can be moved. For example, the burnt front can be moved as long as there is a region to be burnt and a flame's crossed the physical domain in which the solution is sought. On the contrary, in shape recovery the front represents the boundary of an evolving shape. Since the aim is to extract object shapes from a given image, the front should be forced to stop in the vicinity of the desired object boundaries. This is analogous to the back-skeletons used to guide the active contours towards desired shapes. We define the final shape to be the configuration where all the points on the front meet in a sing, thereby bringing the computation to an end.

8.1. Stopping Criterion: An Image-based Speed Term

The goal now is to define a speed function from the image data that can be applied on the propagating front as a stopping criterion. In general the function F can be split into two components: $F = F_A + F_D$. The term F_A , referred to as the advection term, is independent of the moving front's geometry. The front uniformly expands or contracts with speed F_A depending on its sign and is analogous to the collection

form defined in [4]. The second term P_{int} is the part which depends on the geometry of the front, such as its local curvature. This [diffusion] term smooths out the high curvature regions of the front and has the same regularizing effect on the front as the internal deformation energy term in the phase-field models option [10] (see the figure 3 in [4]). We rewrite equation (3.1) by splitting the influence of F as

$$v_t + F_2 \cdot \nabla \psi + \alpha P_2 \cdot \nabla \psi = 0. \quad (3.4)$$

First consider the case where the front moves with a constant speed $\beta \in \mathbb{R}$, $F = F_2$. To that if we add a negative speed term synthesized from the image, such that there are trade to zero some large image gradient locations, we still achieve our goal of halting the front in a ring in the neighborhood of object boundaries. To this end, we define a negative speed P_2 to be

$$P_2(x, y) = \frac{-F_2}{(M_1 - M_2)} \left(|\nabla G_\sigma \circ I|(x, y) - M_2 \right), \quad (3.5)$$

where M_1 and M_2 are the maximum and minimum values of the magnitude of image gradient $|\nabla G_\sigma \circ I|(x, y)$, $(x, y) \in \mathbb{R}^2$. The expression $M_1 + \epsilon$ denotes the image convolved with a Gaussian smoothing filter whose characteristic width is ϵ . Alternatively, we could use a smoothed zero crossing image to synthesize the negative speed function. The zero crossing image is produced by detecting zero crossings on the front line $\nabla^2 G_\sigma \circ I$, which is the original image convolved with a Laplacian of Gaussian filter whose characteristic width is ϵ . The operation of surface with the addition of image based speed is

$$v_t + (P_1 + P_2) \cdot \nabla \psi = 0. \quad (3.6)$$

P_1 is called as extension of F_1 to points away from the boundary $\gamma(\partial\Omega)$, i.e. at points $(x, y) \in (\mathbb{R}^2 - \gamma(\partial\Omega))$, and is equal to F_1 on $\gamma(\partial\Omega)$. We shall return to the topic of

increases sharply. The value of F_1 lies in the range $[-F_2, 0]$ as the value of image gradient varies between M_1 and M_2 . Thus, this argument is clear that the front gradually attains zero speed as it gets closer to the object boundaries and eventually comes to a stop.

In the case when the front moves with a speed that is a function of local curvature, i.e. $F_2 \neq 0$, it is not possible to find an additive speed term from the image that will cause the net speed of the front to approach zero in the neighborhood of a closed shape. Instead, we multiply the speed function $F = F_1 + F_2$ with a quantity h . The term h , which is defined as

$$h(x, y) = \frac{1}{1 + |\nabla G_1 + \nabla G_2|}, \quad (3.4)$$

has values that are close to zero in regions of high image gradients and values that are close to unity in regions with relatively constant intensity. More sophisticated stopping criteria can be implemented by using the curvature dependent "staircase" filter [16]. The modified equation of motion is given by

$$h_t + h[F_1 + F_2] \cdot \nabla \phi = 0. \quad (3.5)$$

3.2. Extension of Image-based Speed Term

We now come to an important point in our discussion. The image-based speed term, h or F_2 or h_2 has meaning only on the boundary $\partial(\Omega)$, i.e. on the level set $\{\phi = 0\}$. This follows from the fact that they were designed to force the propagating level set $\{\phi = 0\}$ to a complete stop in the neighborhood of an object boundary. However, the equation of motion (3.5) is written for the function ϕ , which is made up of infinitely many level sets. In other words, equations (3.5) & (3.6) control the evolution of a family of level sets. Therefore, it is imperative that the net speed used in the evolution equation, has a consistent physical meaning for all the level



Figure 4.1 Hoggan's principle construction

sets, i.e. at every point $(x, y) \in \Omega$. Speed functions such as F_2 which are functions of geometric properties of the surface $x = \phi(x, y)$, can be readily computed at any $(x, y) \in \Omega$. However, F_2 is not such a function. It depends on meaning not from the geometry of ϕ but from the configuration of the level set $\{\phi = 0\}$ in the image plane. Thus our goal is to construct an image-based speed function \tilde{F}_2 that is globally defined. We call it an *extension* of F_2 off the level set $\{\phi = 0\}$ because it extends the meaning of F_2 to other level sets. Note that the level set $\{\phi = 0\}$ lies in the image plane and therefore \tilde{F}_2 must equal F_2 on $\{\phi = 0\}$. The same argument applies to the coefficient b_2 .

If the level curves are moving with a constant speed, i.e. $\dot{F}_2 = 0$, then at any time t a typical level set $\{\phi = C\}$, $C \in \mathbb{R}$ is a distance C away from the level set $\{\phi = 0\}$ (see Figure 4.1). Observe that the above statement is a rephrased version of Hoggan's principle which, from a geometrical viewpoint, stipulates that the position of a front propagating with unit speed at a given time t should consist of only the set of points located a distance t away from the initial front. On the other hand, if $\dot{F}_2 \neq 0$, the level sets will violate the property that they are a constant distance away from each other. However, they will never collide and cross each other if the speed



Figure 4.3: Extension of stage-based speed (costs) to other level sets

function $P = P_A + P_B$ is continuous (see [14]). With the above smoothness used we state the following

Remark 4.3.1. Any rational (stage-based) speed function that is used in the system of motion equations for the function ϕ should not cause the level sets to collide and cross each other during the evolutionary process.

Recall that the function $\phi(x, t)$ has been extended to $d(x)$, where $d(x)$ is the signed distance from a point $x \in D$ to the boundary $\phi(t)$. Since we cannot introduce any geometry coming in the function $P_1(h_1)$ at points away from the level set $\{y = 0\}$, we look for a meaning that is consistent with property (S 2.1). Therefore, the question to ask is: what is the value of P_1 (or h_1) at a point (x, y) lying on a level set $\{y = C\}$? We answer this question in the following construction (see Figure 4.3).

Definition 4.3.1. The value of $P_1(h_1)$ at a point P lying on a level set $\{y = C\}$ is exactly the value of $P_1(h_1)$ at a point Q such that point Q is a distance C away from P and lies on the level set $\{y = 0\}$.

It is easy to see that P_1 reduces to P_A on $\{y = 0\}$. We use the same construction to determine the value of h_1 at a point (x, y) lying on some level set $\{y = C\}$.

Note that if the definition of a speed function adheres to constraints 4.2.1, then it will also be consistent with the property 4.2.3. Thus, having ascribed the intent of pseudo-differential equations (4.2) & (4.3) in the context of shape modeling, we can now define difference schemes to solve them numerically. Since it can develop corners and sharp gradients, numerical schemes based on finite hyperbolic conservation laws are used to produce stable speed solutions. Moreover, the equations of motion can be solved on a uniform mesh and the level sets can be moved without their explicit construction.

4.1. Motion Equation in 2D

In this section, we show that the moving shapes of three-dimensional objects can be recovered using a scheme that is analogous to the one described above. We begin by writing the equation of motion for a moving surface that represents front propagation in higher dimension. Let the surface be closed, non-intersecting, and the level set $\{\phi = 0\}$ of a function $\phi(x, t) \in \mathbb{R}^n$. The function $\phi(x, t = 0) = \pm d(x)$, where $d(x)$ is the signed distance function from the point x to the initial surface, plus sign is used if x is outside the initial surface and minus sign if it is inside. Each level set of ϕ moves along its gradient field with speed $F(x)$. Following the derivation in chapter 3, we arrive at the equation of motion for the moving surface, namely

$$\phi_t + F(x)|\nabla\phi| = 0, \quad (4.4)$$

where F is either mean or Gaussian curvature of the surface. Recall that the surface at any time t is simply the level set $\{\phi = 0\}$.

In this work, we only address the issue of recovering surface shapes from 2D images. The input data constitutes a set of MRH (Magnetic Resonance Imaging) or CT (Computed Tomography) images. A stack of such images represents an image function $I(x, y, z)$. Our next step is to define an image-based speed term which will

near the level set, $\{x_i = 0\}$ to stop near the 3D object boundary. To this end, we apply on the front the speed equal to the product of $F(x)$ and a term k_2 . The term k_2 , which is defined as

$$k_2(x, y, z) = \frac{1}{1 + \left(\nabla \phi_{\text{3D}}^2 + I(x, y, z) \right)}, \quad (4.7)$$

low values closer to zero in the vicinity of object boundaries and values that are close to unity in regions of relatively constant image intensity. To introduce a consistent meaning to the image-based speed term at all the level-sets, we construct an xy -based term k_2 as a function similar to the construction (4.3). Therefore, the modified equation of motion that we solve to extract complex surface shapes from 3D images is

$$\partial_t x + k_2 F(x) \nabla \phi = 0. \quad (4.8)$$

CHAPTER 1 NUMERICAL SOLUTION

The equation (1.1) poses an initial value problem. It is written here as

$$\phi_t + (\phi_x^2 + \phi_y^2)^{1/2} = cV \left\{ \frac{\nabla \phi}{|\nabla \phi|} \right\} \quad (1.1)$$

with $\phi(x, y, t = 0) = \frac{1}{2}$ defined over $[a, a] \times [a, a]$. As shown in Section [26], for $c > 0$, the parabolic right hand side diffuses sharp gradients and forces ϕ to stay smooth at all values of t . For $c = 0$, the boundary moves with unit speed, and a corner must develop from smooth initial data. Once a corner develops, it is not clear how to propagate the front in the normal direction, since the direction is not defined at the corner. A variety of "weak" solutions which propagate the curve beyond the occurrence of a cuspiness is possible. Of all such weak solutions, one is interested in the one that is the limit of smooth solutions as $c \rightarrow 0$. This particular weak solution can be selected with the help of a so-called "entropy condition", see [26]. If the front is moved as a burning flame, then one is looking for front of step form. Thus, approximations to the spatial derivatives are sought that do not artificially smooth sharp corners and which pick out the correct entropy solution when singularities develop. The solution given in [10], [25] are motivated by the fact that the entropy condition for propagating fronts is identical to the one for hyperbolic conservation laws, where stable, consistent, entropy satisfying algorithms have a soft history

3.1. Entropy satisfying Upwind Numerical Scheme

In this section, almost without a change, we present the arguments discussed in Section 26. For complete details of the following scheme, we refer the reader to Gassner and Reissig [34]–[35].

First, consider the one-dimensional version of the level set equations, with constant normal velocity $F_N = 1$. Then the equation (3.2) becomes a standard Hamilton-Jacobi equation

$$d_t + H(d_t) = 0, \quad (3.3)$$

where $H(d_t) = -(\phi_t^2)^{1/2}$, and with a given initial value of ϕ . Let $u = d_t$. Taking the derivative with respect to x , equation (3.3) becomes

$$u_t + (H(u))_x = 0, \quad (3.4)$$

where $H(u) = -(\phi^2)^{1/2}$. Equation (3.4) is a scalar hyperbolic conservation law in one space variable. Solutions can develop discontinuous jumps known as shock waves with smooth initial data. In order to make sense of the solution after shock form, an integral version of the conservation law which admits discontinuous solutions is needed. Both sides of equation (3.4) are integrated on an arbitrary interval $[a, b]$ to produce

$$\frac{d}{dt} \int_a^b u(x, t) dx = H(u(a, t)) - H(u(b, t)) \quad (3.5)$$

It is known as a weak solution of the conservation law if it satisfies the above integral equation. Note that u need not be differentiable to satisfy the integral form of the conservation law.

When will a numerical algorithm approximate the correct entropy satisfying solution to equation (3.4)? The answer is found in the following definition.

Definition 3.1.1. Let n_i^j be the value of n at a mesh point x_i at time t_j . A three-point difference scheme is said to be in conservative form if there exists a function $g(n, u_i)$ such that the scheme can be written in the form

$$\frac{n_i^{j+1} - n_i^j}{\Delta t} = -\frac{g(n_i^j, u_{i+1}^j) - g(n_{i-1}^j, u_i^j)}{\Delta x}, \quad (3.4)$$

where $g(n, u) = h^2 \eta(u)$.

For definition in terms of the scheme must approximate the hyperbolic conservation law, subject to the consistency requirement, $g(n, u) = h^2 \eta(u)$. In order to guarantee that the scheme picks out the correct entropy satisfying weak solutions, monotonicity is required, i.e., that n_i^{j+1} be an increasing function of the arguments n_{i-1}^j , n_i^j and n_{i+1}^j . The next fact is: A conservative, monotonic scheme produces a solution that satisfies the entropy condition. Equation (3.5) is a scheme for the slope α , which must be regarded here as scheme for g itself. First write equation (3.5) with a forward difference in time as

$$n_i^{j+1} - n_i^j = \Delta t \eta(\alpha) \quad (3.6)$$

Since the numerical flux function g approximates $h^2 \eta$, the solution to equation (3.6) may be approximated by

$$\begin{aligned} n_i^{j+1} &= n_i^j + \Delta t g(n_{i-1}^j, n_i^j, n_{i+1}^j) \\ &= n_i^j + \Delta t g(D_-^j(n_i^j), D_+^j(n_i^j)), \end{aligned} \quad (3.7)$$

where g is an appropriate numerical flux function and the standard definitions of the forward and the backward difference operators have been used, namely

$$\begin{aligned} D_+^j(n_i^j) &= \frac{n_{i+1}^j - n_i^j}{\Delta x}, \\ D_-^j(n_i^j) &= \frac{n_i^j - n_{i-1}^j}{\Delta x} \end{aligned} \quad (3.8)$$

Finally, an appropriate numerical flux function g is required. In the special case where $R[\phi]$ may be written as a function of ϕ^2 , i.e., $R[\phi] = \tilde{f}(\phi^2)$ for some function \tilde{f} , one can use the flux function g as function given in [30]:

$$\begin{aligned} g(\mathbf{R}_L, \mathbf{R}_R, \mathbf{U}_L, \mathbf{U}_R) &= f(\mathbf{U}_L^* \cdot \mathbf{R}_L, \mathbf{U}_L^* \cdot \mathbf{U}_R) \\ &= \tilde{f}(\max\{\mathbf{U}_L^* \cdot \mathbf{U}_L, \mathbf{U}_L^* \cdot \mathbf{U}_R\}^2 + (\max\{\mathbf{U}_L^* \cdot \mathbf{U}_L, \mathbf{U}_L^* \cdot \mathbf{U}_R\})^2) \end{aligned} \quad (3.17)$$

The conservative numerical scheme is applied instead as that it differs only in the definition of propagating characteristics. This is important, since it requires bound any variables on the scale of a finite-word computational bus. An applied scheme numerically different in the outward facing direction at the bus ends if the boundary is expanding, then information flows out. In the case where $P_A = 1$ so that $\dot{\phi}(x^*) = -|x^*|^{P_A}$, equation (3.17) reduces to:

$$\mathcal{F}_L^{out} = \mathcal{F}_L^* - \Delta\mathbf{R}(\max\{\mathbf{U}_L \cdot \mathbf{U}_L, \mathbf{U}_L \cdot \mathbf{U}_R\}^2 + (\max\{\mathbf{U}_L^* \cdot \mathbf{U}_L, \mathbf{U}_L^* \cdot \mathbf{U}_R\})^2)^{1/2} \quad (3.18)$$

This algorithm produces the correct entropy-satisfying weak solution to the propagating front problem.

The above discussion can be extended to problems in more than one space dimension [see Götts and LeChien [20]]. Recall that the original intent was to solve equations (1.2) and (1.4). In two dimensions, the scheme given in equation (3.18) is extended by differencing in each direction to produce the following numerical scheme for equation (1.2):

$$\begin{aligned} \mathcal{F}_L^{out} &= \mathcal{F}_L^* - \Delta t P_A + (P_A)_{L,R}(\max\{\mathbf{U}_L \cdot \mathbf{U}_{L,R}, \mathbf{U}_L^* \cdot \mathbf{U}_R\}^2 + (\max\{\mathbf{U}_L^* \cdot \mathbf{U}_{L,R}, \mathbf{U}_L^* \cdot \mathbf{U}_R\}^2 \\ &\quad + (\max\{\mathbf{U}_L^* \cdot \mathbf{U}_{L,R}, \mathbf{U}_L^* \cdot \mathbf{U}_R\}^2 + (\max\{\mathbf{U}_L^* \cdot \mathbf{U}_{L,R}, \mathbf{U}_L^* \cdot \mathbf{U}_R\})^2)^{1/2} \end{aligned} \quad (3.19)$$

Similarly, the numerical scheme for equation (1.4) is,

$$\begin{aligned} \mathcal{F}_L^{out} &= \mathcal{F}_L^* - \Delta t F_d(\mathbf{U}_L)_{L,R}(\max\{\mathbf{U}_L \cdot \mathbf{U}_{L,R}, \mathbf{U}_L^* \cdot \mathbf{U}_R\}^2 + (\max\{\mathbf{U}_L^* \cdot \mathbf{U}_{L,R}, \mathbf{U}_L^* \cdot \mathbf{U}_R\}^2 \\ &\quad + (\max\{\mathbf{U}_L^* \cdot \mathbf{U}_{L,R}, \mathbf{U}_L^* \cdot \mathbf{U}_R\})^2 + (\max\{\mathbf{U}_L^* \cdot \mathbf{U}_{L,R}, \mathbf{U}_L^* \cdot \mathbf{U}_R\})^2)^{1/2} - \Delta t F_{in}(\mathbf{U}_L) \cdot \mathbf{U}_L \end{aligned} \quad (3.20)$$

The second term $F_{\text{edge}}[|\nabla\psi|]$ is not approximated in the above equation, one may use a straightforward central difference approximation to this term

3.1. Image-based Update Algorithm

In this section we first consider a straightforward approach to solve the equations (2.17) & (2.18) and subsequently show that the same complexity of our algorithm can be expressed by applying the update procedure to a small set of points in the domain. In other words, we propagate the front by solving the governing equations at a small number of points. These points lie on a narrow band around the level set $\{\psi = 0\}$.

It should be observed that by updating the level set function on a grid, we are moving the level set without constructing them explicitly. A particular level set $\{\psi = 0\}$, which models the shape of interest, is moved by a step in the vicinity of desired object boundaries. This task is accomplished by applying an image-based speed term on the zero set equation (4.2). An unknown function is constructed to simulate a constant velocity in the image-based speed term at points away from the zero set (see discussion in section 4). Therefore, a straightforward algorithm consists of advancing from one time step to the next as follows

Algorithm 1

1. At each grid point (x_k, y_k) , where x_k and y_k are step size in x -axis coordinate directions, the evaluation of image based speed term is computed. This is done in accordance with the procedure described in previous section, i.e., by searching for a point q which lies on the level set $\{\psi = 0\}$, and is closest to the point (x_k, y_k) . The value of image-based speed term at the current point is simply the value at the point q .

1. With the value of extended speed term $\{d_{\text{ext}}^i\}_{k=0}^n$ and d_{ext}^0 , calculate d_{ext}^{i+1} using the spread, finite difference scheme given in equation (5.12).
2. Construct an approximation for the level set $\{\phi = 0\}$ from d_{ext}^{i+1} . This is required to translate the current position of the front in the image plane. A piecewise-linear approximation for the level $\gamma[\phi]$ is constructed as follows. Given a cell $C[x_{i,j}]$, if $\text{max}\{\theta_{k,0}, \theta_{k+1,0}, \theta_{k+2,0}, \theta_{k+3,0}\} < \pi$ or $\text{min}\{\theta_{k,0}, \theta_{k+1,0}, \theta_{k+2,0}, \theta_{k+3,0}\} > \pi$, then $C[x_{i,j}] \notin \gamma[\phi]$ and is ignored, else, the entry as it exits points where $\phi = 0$ is found by linear interpolation. This provides two nodes on $\gamma[\phi]$ and thus, one of the line segments which form the approximation to $\gamma[\phi]$. The collection of all such line segments constitutes the approximation to the level set $\{\phi = 0\}$, which is used for future evaluation of the image-based speed term in the update equation (5.12).
4. Replace n by $n + 1$ and return to step 1.

It is easy to see from the above algorithm that the most expensive step is the computation of the extension for image-based speed term. This is because at each grid point, we must search for the closest point $\{x_{k,0}\}$ on the level set $\{\phi = 0\}$. However, if $P_0 = 0$, then the stability requirement for the explicit method for solving our level set equation is $\Delta t = O(\Delta x)$. For the full equation (5.12), the stability requirement is $\Delta t = O(\Delta x^2)$. This could potentially force a very small time step for fine grids. These two effects, collectively and compounded, make the computations exceedingly slow. A marginal improvement in performance can be achieved by evaluating the extension only once every 4 iterations. In other words, we take 4 steps (4 times without recomputing the image-based speed $\{d_{\text{ext}}^i\}$). Alternatively, we could down sample the image and perform our calculations at a lower resolution. On the other hand, we run the risk of losing accuracy while computing on a coarse grid. Our approach is to



Figure 3.1: A narrow band of width f around the level set $\{\phi = 0\}$

which run level set algorithm as a neighborhood traversal. Instead, we improve the time complexity of our algorithm by updating the level set function ϕ only at a small set of points on the grid.

The basic idea behind this new scheme stems from the observation that the front can be moved by updating the level set function at a small set of points in the neighborhood of the zero set instead of updating ϕ at all the points on the grid. In Figure (3.1) the bold curve depicts the level set $\{\phi = 0\}$ and the shaded region around it is the narrow band. The narrow band is bounded on either side by two curves which are a distance f apart, i.e. the two curves are the level sets $\{\phi = \pm f/2\}$. The value of f determines the number of grid points that fall within the narrow band. Hence, during a given time step the value of ϕ_{ij} is not updated at points lying outside the narrow band; the level sets $\{\phi \geq f/2\}$ remain stationary. The zero set which has made recent solid/collision with the boundary of the narrow band. Which boundary the front collides with depends on whether it is propagating inward or outward, i.e. which way it is about to move past the narrow band's boundary.

Therefore, as a consequence of our update strategy, the front can be moved through a maximum distance of $R/3$, either upward or downward, at which point we must rebuild an appropriate narrow band. We recalculate the ϕ function by finding the current narrow configuration, i.e., $\{p = 0\}$, as the filled curve $\phi(\tilde{p})$ (see section 5). Note that the recalculation procedure must account for the case where $\{p = 0\}$ changes topology. This procedure will reduce the recalculation of ϕ function by extending the recurrence introduced as a result of our update algorithm. Once a new ϕ function is defined on the grid, we can create a new narrow band around the zero set, and go through another set of, say, L iterations in order to move the front ahead by a distance up to $l/3$. The value of l is set as the number of time steps required to move the front by a distance roughly equal to $R/3$. This choice depends on some system parameters. Thus, a faster algorithm for slope recovery consists of the following steps:

Algorithm 2

1. Set the division number $m = 3$ and go to step 2.
2. At each grid point (i, j) (reg. inside the narrow band), compute the estimate \hat{h}_i^m of slope-based speed term.
3. With the above value of estimated speed term $\{\hat{h}_i^m\}_{i,j}$, and $\hat{h}_{i,j}^0$, calculate $\hat{v}_{i,j}^{m+1}$ using the speed, flux difference scheme given in equation (5.11).
4. Construct a polynomial approximation for the level set $\{p = 0\}$ from $\hat{v}_{i,j}^{m+1}$. A contour tracing procedure is used to obtain a polynomial approximation. Given a cell (i, j) which contains $\phi(\tilde{p})$, the procedure traces the contour by visiting the neighboring cells in order to find the next cell which contains $\phi(\tilde{p})$. Once such a cell is found, the process is repeated until the contour closes on itself.

The set of nodes visited during this tracing process constitutes the polygonal approximation to $\gamma(\Omega)$. In general, to collect all the closed contours, the above tracing procedure is started at a new, as yet unvisited, cell which contains the level set $\{\phi = 0\}$. A polygonal approximation is required in step 2 for the evaluation of integrated speed term and more importantly, in step 4 for evaluating the ϕ function.

3. Increment m by one. If the value of m equals K_{max} in step 1, then go to step 2.
4. Compute the value of signed distance function ϕ by tracing the polygonal approximation of $\{\phi = 0\}$ in the latest contour $\gamma(\Omega)$. An extended notion, a more general method of reinitialization is required when $\{\phi = 0\}$ changes topology. Go to step 1.

The issue of boundary conditions needs to be addressed at this time. In our original method, the level set function ϕ was initialized and updated on a square grid. To update ϕ_0 at a boundary point (x, y) , it is necessary to specify the boundary conditions in order to numerically approximate the derivatives. We have chosen free out boundary conditions, i.e., if a derivative computation at a given boundary point $[x, y]$ involves a point that falls outside the computational domain, then that derivative is set to zero. In our new approach, since we only update ϕ at points lying on the curve itself, the issue of specifying boundary conditions for points lying on the edge of the level becomes pertinent. The issue of level propagation in the context of image recovery is merely rapid advancement of the front. Therefore, at the expense of a little inaccuracy in the new set configuration, we ignore the sensitive issue of specifying correct boundary conditions. In our implementation, derivatives are evaluated normally even if it involves a point that does not belong

in the narrow band. On the other hand, in applications such as crystal growth, and flame propagation, accurate specification of boundary conditions is important.

We now show that the new faster approach provides a correct approximation to the propagating front problem. In figure 5 (i) (j), we show the result of applying narrow band update algorithms to a star shaped front propagating with speed $F = -K$, where K is the curvature as in equation (2.6). The calculation was done on a unit disc with 60 points in radial direction, and a time step of $\Delta t = 0.001$ was employed. The width of the narrow band has been set to $\delta = 0.075$ and the ψ function was recomputed once every $|F| = 40$ time steps. In figure 5.2(a), we show the initial curve along with the level sets ($\psi = 0.0001$) where $i \in [1, 5]$. After 40 narrow band updates (figure 5.2(b)), only the level sets ($\psi < 0.001$) move and the rest remain stationary. We note the consistency between the level sets lying on either side of the narrow band, making the reinitialization step necessary in order to correct the smoothing of the ψ function. Following the reinitialization step, another 40 update steps are applied (figure 5.2(c)), which “diffuses” the high curvature regions of the front even further. In subsequent figures, the results of repeatedly applying the same strategy are shown. Finally, in figure 5.2(f), the peaks and troughs on the front get completely diffused, and it attains a smooth circular configuration after 4 reinitialization steps and a total of 200 time steps.

We conclude this section by highlighting some advantages of the narrow band update scheme. The first is the obvious improvement in time complexity as a result of considering only a fraction of the total number of grid points for updates. The second is related to the issue of computing the value of image-based speed terms at points away from the zero set, i.e., the interface. In our original method, the extension of image-based speed terms could be unreliable at points that are relatively far from the zero set. In addition, when the front is a circular disc it is not clear



(a) Initial density



(b) After 10 iterations



(c) After 50 iterations



(d) After 100 iterations



(e) After 150 iterations



(f) After 200 iterations

Figure 5.1: Numerical solution algorithm applied to a star-shaped front going with speed $P = -0.5$. Calculations were done on a 50×45 grid with a time step $\Delta t = 0.002$. ϕ was recomputed after every 10 time iterations.

what value of image-based speed term to use at its center. Here, the fact that the center of a circle is equidistant from all the points lying on it causes no ambiguity in the choice of the closest distance point on the circular level. Clearly, this effect is undesirable. The surface-based update algorithm does not require the image-based speed term's evaluation at its wrap points. Therefore, the solution is very accurate and reliable.

3.1.3. Numerical Issues in 2D

In this section, we provide some details of the numerical method based upon the Eulerian formulation [46] that is used to solve the equations (4.4). To begin, the discussion in the previous section can be readily extended to 2D by differentiating in each space dimension to produce the following numerical scheme to approximate the surface motion:

$$\begin{aligned} \psi_{i,j,k}^{(n+1)} = & \psi_{i,j,k}^{(n)} - \Delta t (\dot{h}_0)_{i,j,k} \{ (\max(D_x \psi_{i,j,k}, 0)) \bar{U}^2 + (\max(D_x^* \psi_{i,j,k}, 0)) \bar{U}^2 \\ & + (\max(D_y \psi_{i,j,k}, 0)) \bar{U}^2 + (\max(D_y^* \psi_{i,j,k}, 0)) \bar{U}^2 \\ & + (\max(D_z \psi_{i,j,k}, 0)) \bar{U}^2 + (\max(D_z^* \psi_{i,j,k}, 0)) \bar{U}^2 \}^{1/2} - \Delta t (\rho_0 K_0 (\dot{h}_0) / \nabla \psi) \} \end{aligned} \quad (5.13)$$

Note that the level surfaces are moving with speed $\bar{U} = \dot{h}_0(t) = \dot{h}(R)$, where R is the outer radius.

In the context of 2D shape recovery, we move the level surfaces by updating the ψ function on a three dimensional grid according to the above update equation. The level surface $\{\psi = 0\}$, which models the shape of interest, is lifted to a step on the vicinity of desired object boundaries by applying an image-based speed coefficient on it. In order to be compute the value of image-based speed coefficient at points away from the boundary, an influence is constructed. More specifically, the shape recovery procedure in 2D consists of the following steps:

Algorithm 3

- 1 At each grid point (x_{ij}, y_{ij}, z_{ij}) the solution of source-based speed on \mathcal{B} next to \mathcal{B}_i is computed
- 2 With the value of estimated speed coefficient $\hat{Q}(x_{ij})$ and $\hat{Q}_{\mathcal{B}_i}$, calculate $\hat{Q}_{\mathcal{B}_i}^{(i)}$ using the spread, finite difference scheme given in equation (3.14)
- 3 Construct an approximation for the level surface $\{\phi = 0\}$ from $\hat{Q}_{\mathcal{B}_i}^{(i)}$. This is done in the following fashion. Across every row vertices of a given cell $\mathcal{C}(x_{j-1}, x_j)$, the value of $\hat{\phi}$ function is sampled. If the $\hat{\phi}$ function changes sign across a given edge, the exact point at which $\hat{\phi} = 0$ is found by linear interpolation. Note that at each cell, the $\hat{\phi}$ function can change sign across a number of edges that lies between 1 and 6. The collection of all such points constitutes the approximation to the level surface $\{\phi = 0\}$, which is used for the evaluation of \hat{L}_i during the next time step
- 4 Replace n by $n + 1$ and return to step 1

In the previous section, it has been established that the extension calculation is very expensive. This observation has motivated the narrow band approach for front propagation. The situation in 2D is even worse due to its entire domain. In addition, there is a large degree of repetition in the new set due to the lack of an organized method in which to populate the new set. By organized method we mean a 3D analog of a 2D contour tracing procedure. This limitation adds to the cost of the new set and subsequently to the time complexity of extension calculation. In 3D, we alleviated this situation by adopting a narrow band update strategy, but implementing a similar scheme in 2D could prove overly cumbersome. Therefore, we exploit

the inherent data independence that is present in our grid calculation, and design a parallel implementation to solve the level surfaces. The parallel implementation was done on a KSR (Kendall Square Research) shared memory parallel machine. For a grid with 10^5 points, we have used N processors and realized a substantial improvement in the running time of our 3D shape recovery algorithm. Steps 1 & 2 of the above algorithm are done in parallel and step 3 is executed in serial mode.

CHAPTER 4 EXPERIMENTAL RESULTS

In this chapter we present some shape recovery results that were obtained by applying the narrow band level set algorithm to range data. Subsequently we present some results in 2D. First, in 2D, given an image, our method requires the user to provide an initial contour $\gamma(0)$. As we shall see, there is absolutely no restriction on where the initial contour can be placed in the image plane as long as it is made a desired shape to recover all the contained shapes. The feature is of pronounced importance in the context of automatic shape recovery. Our front moves the object boundaries by either propagating outward in the normal direction or propagating inward in the negative normal direction. This shows it made of the form of initialization. Note that after the specification of initial shape of $\gamma(0)$, our algorithm does not require any further user interaction.

The initial value of the function $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ is computed from $\gamma(0)$. We first determine the level set function ϕ on the image plane and denote $\phi_{i,j}$ as the value of ϕ at a grid point $(i\Delta x, j\Delta y)$, where Δx and Δy are step sizes in either coordinate directions. In our implementation, there we usually work with $2^p \times 2^p$ images, the computational domain is a square with $\Delta x = \Delta y = h$. We define the distance from a point (i,j) to the initial curve to be the shortest distance from (i,j) to $\gamma(0)$. The magnitude of $\phi_{i,j}$ is set to this value. We set the plus sign if (i,j) is outside $\gamma(0)$ and minus sign if (i,j) is inside. Once the value of $\phi_{i,j}$ is computed at least 1×1 by following the above procedure, we use the update equations from the previous chapter to move the front.

5.1. 2D Phase Boundary Results

We now present our shape recovery results in 2D. We first consider a 256×256 CT (computed tomography) image of an abdominal section, shown in figure 6.1(a). Our intention is to recover the shape of the liver in this particular slice. The function ϕ has been discretised on a 128×128 mesh, i.e., calculations are performed at every second pixel. In figure 6.1(b), we show the closed contour that the user places inside the desired shape at time $t = 0$. The function ϕ is then made to propagate on the grid at discrete with speed $F = k_f \cdot (1 - \phi) \cdot 0.05[N]$. We have employed the narrow band update algorithm to move the front. The time step size was set to $\Delta t = 0.005$ and the ϕ function was recomputed after every 10 time steps. Figure 6.1(c) shows the image-based speed term which is implemented according to equation (5.4). This term when applied on the propagating front, acts as a stopping criterion, thereby ensuring the front to come to a rest in the neighbourhood of a closed shape. Note that in figure 6.1(c), $k_f(x, y)$ values lying in the interval $[0, 1]$ have been mapped into the interval $[0, 200]$. In figures 6.2(a) through 6.2(d) we depict the configurations of the level set $\{\phi = 0\}$ at four intermediate-time instants. The final result is achieved after 470 time iterations and is shown in figure 6.2(e). It should be noted that our method does not require that the initial contour be placed close to the object boundary. In addition, observe how the front overflows all the isolated spurious edges present inside the shape (see figure 6.1(b)) and settles in the neighbourhood of edges which correspond to the true shape. This feature is a consequence of (5.4) component in the speed which reflects regions of high curvature on the front and forces it to attain a smooth shape.

As mentioned in section 3, smoothness of the front can be controlled by choosing an appropriate curvature component in the speed function $F = 1 - \alpha k^2$. The algorithm

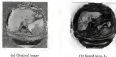


Figure 4.1: Image-based speed tests $\hat{H}_2(x, y) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-y)^2}{2\sigma^2}}$ with $\sigma = 3$ Hz, synthesized from the CT image.

of our next experiment, is to demonstrate overfitting control in the context of shape recovery. In figures 4.3(a) through 4.3(f), we show the results of applying our narrow-band shape recovery algorithm to an image consisting of three synthetic shape perturbations, was performed by drawing a curve enclosing each one of the three shapes. We compute the signed distance function $\phi(x, y)$ from these curves. The level sets are then made by propagating with speed $F = \hat{H}_2(y)$ from K . First, as shown in figure 4.3(a), we perform shape recovery with the value of $\epsilon = 0.05$. The process is repeated with different values of ϵ , 0.10 in figure 4.3(b) and 0.70 in figure 4.3(f). Clearly, with every increment in the value of ϵ , the level set $\{\phi = 0\}$ attains a configuration that is relatively smoother. This is analogous to the controlled overfitting provided by the plate-mechanism system [41–47].

In our third experiment we recover the complicated structure of an arterial tree. The real image has been obtained by clipping a portion of a digital subtraction angiogram. This is an example of a shape with branches and irregular protrusions. In this experiment we compare the performance of our scheme with the conventional



(a) Initialization



(b) After 100 iterations



(c) After 225 iterations



(d) After 350 iterations



(e) After 475 iterations



(f) After 600 iterations

Figure 4.3: Recovery of the brain shape from a CT image of an abdominal section. Without facial computations (see [10] for a 2D + 2D grid - the front was made to propagate with speed $F = \lambda_2(-1.8 - 0.005k)$ and the time step Δt was set to 0.0005 - ϕ was reinitialized near every 50 time steps.



Figure 4.4: Sometimes control in shape recovery can be achieved by varying the curvature component in the speed $F = k_F(\cdot) (1 - \epsilon(k))$

model is being the low distance of the later interface. We first attempt to reconstruct the complex internal structure using a snake model with reflection forces [3]. In figures 4.4(a) through 4.4(c), we show a sequence of pictures depicting the snake configurations in the image. We present the final equilibrium state of the snake in figures 4.4(e)–4.4(g), for 3 different initializations, not better than the preceding. In all these cases the active-contour model, even after 1000 time iterations, barely recovers the main idea of the image and completely fails to account for the branches. Two potential limitations of the snake model immediately come into light. The first is the dependence of final result on the initial configuration. This is due to the existence of multiple local minima in the (parametric) energy functional which the minimization procedure explicitly minimizes. The second failure is the inability of snake model to obtain a stable shape with perimeter. Observe here in the third case, despite a good initialization (figure 4.4(g)), the snake “snaps” back into a relatively “spherical” configuration in figure 4.4(h). The discrepancy stems from snake’s any-length (elasticity) and curvature (rigidity) rescaling nature. Snake prefers regular shapes (circles, shapes with perimeter have very high dilating energy). Now

We apply our level-set algorithm to reconstruct the screw shape. After initialization in Figure 4.5(a), the level is made to propagate in the normal direction. By using the narrow-band update scheme with a band width of $d = 0.05$ to move the level, it can be seen that, in subsequent frames the level finally “fills” into the boundary and finally in 4.5(f) it completely reconstructs the complex tree structure. The advantage of our scheme are quite apparent from this example. Since our level advancement process does not involve optimization of any quantity, the shape recovery results we obtain are independent of initialization. In addition, a single instance of our shape model “grows” boundary and covers all the connected components of a given shape. In Figure 4.5(a)-(f) we plot the other level sets to elucidate the process of estimating the shape-based speed function to provide away from the zero set. All calculations were carried out on a 512×512 grid and a time step $\Delta t = 0.001$ was used.

In the next experiment, we depict a situation when the level undergoes a topological transformation to reconstruct the crocheted shapes in an image. The image shown in Figure 4.5(g) consists of three distinct shapes. Initial curve is placed in such a way that it overlaps all the objects. The level is then advanced in the direction of negative normal. Alternately, we could perform the initialization by placing a curve in each one of the individual shapes and propagating them in the normal direction. We choose the former option. The level set ($\phi = 0$) first wraps itself tightly around the objects (see Figure 4.5(g) & 4.5(h)) and subsequently splits into four separate closed curves (Figure 4.5(i)). While the first three closed segments of $\phi = 0$ enclose the three distinct shapes, the one in the middle (see Figure 4.5(i)), which it does not enclose any object, eventually disappears. Figure 4.5(j) shows the final result. Again it should be noted that a single instance of our shape model dynamically splits into three instances to represent each object. To show the working of our algorithm, in

Figure 4.8(a)-(c) we show the level sets $\{\phi = \text{int}(\mathcal{C})\}$, $t \in [-0.5, +0.5]$, with $\mathcal{C} = \mathbb{R}^2$. The function ϕ was discretized on a 100×100 grid and Δt was set to 0.001.

Lastly, we show that our approach can also be used to recover shapes with holes. We do this by applying our method to extract the shapes of hand-written characters. The characters "A" and "B" in the figure 4.9(a) are examples of shapes with holes. Recovery of hand-written character shapes finds applications in the area of character recognition. These shapes can also serve as an input stage in algorithms computing medial axis transforms [30, 31] or skeletons [32, 33]. The intermediate here is the manner in which the outer and inner boundaries of a given shape can be recovered without separate calculations. In figure 4.8(a), we show the initial contour which encloses all the character. This contour is then made to propagate inward with a constant speed. Figure 4.8(b) shows an intermediate stage in the level set evolution and in 4.8(c), the level sets split into separate contours. The calculation comes to a halt when in figure 4.8(d), the level set $\{\phi = 0\}$ recovers the outer boundaries of four separate characters. Unlike the characters "A" & "B" for which we need to extract the inner boundary for their complete shape description, for the characters "D" & "E", the inner boundary completely describes their shape. In the second stage of our computation, we reset the zero set configuration in figure 4.8(d) as an initial state, and propagate all the fronts inward by incrementally reducing the image-based speed term. This causes the zero set to move into the character shapes as shown in figure 4.8(e). Finally in figure 4.8(f), the zero set comes to stop in the neighborhood of holes that are present in the shapes of characters "A" & "B", thereby achieving a complete shape recovery. The calculations for this experiment were done on a 150×150 grid and the time step Δt was set to 0.0005.

4.1. Experimental Results in 2D

In this section, we present some front propagation and shape recovery examples in 2D. The first two experiments demonstrate the basic surface motion scheme in 2D with a constant and curvature-dependent speed. In situations the level set $\{\phi = 0\}$, we use a two-dimensional algorithm. Given a function $\phi_{k+1/2}$ on a grid, this algorithm first marks off all the points with negative ϕ values (here we assume that the points lying inside the surface have negative ϕ values and the points lying outside have positive ϕ values). A two-dimensional procedure is then applied to these points. The boundary points that lie on the boundary of the initialization constitute our approximation to the level surface $\{\phi = 0\}$. Using these surface triangles the level set $\{\phi = 0\}$, which forms our 2D surface model, is rendered as a shaded object.

In our last 2D experiment, we evolve a toroidal surface with constant speed $F = 1$. The computational box is a cubical one which extends from $[-0.5, 0.5] \times [-0.5, 0.5]$ in all three coordinate directions. Initial value of ϕ is defined according to the following equation

$$\phi(x, y, z) = 0.005 - \frac{1}{2}(x^2 + y^2)^{0.75} - 0.5(z^2 - x^2). \quad (5.1)$$

The set of all points (x, y, z) at which the value of $\phi = 0$, lie on the surface of the torus (see figure 4.10(a)). We discretize this ϕ on a $50 \times 50 \times 50$ grid and follow a sorted set of toroidal shapes by updating it on the grid according to the equation (5.14). Figures 4.10(b), 4.10(c) depict three intermediate stages wherein the mean value gradually becomes smaller until in figure 4.10(d) the inner surface collapses to zero, annulus splits, and topology changes. The surface at this stage looks like a deformed sphere and will asymptotically attain a spherical configuration. In figure 4.10(e), the level set $\{\phi = 0\}$ coincides with the edge of the computational box (forming the closing of the toroidal surface).

In our next experiment, we collapse a “pushed” superquadrupole shape under its mean curvature. The optimization was done using the implicit function form of a superquadrupole, namely

$$\psi(x, y, z) = \left[\left(\left(\frac{x}{a_1 b_1} \right)^{2n_1} + \left(\frac{y}{b_1 c_1} \right)^{2n_2} \right)^{1/n_1} + \left(\frac{z}{c_1 d_1} \right)^{2n_3} \right]^{1/n_2} - 1/2, \quad (4.1)$$

where $a_i, b_i \geq 0$ are “squariness” parameters. By varying the values of n_1 and n_2 , varied interesting shapes can be generated. Flat beveled shapes are produced when either n_1 or $n_2 = 1$ and pushed shapes are produced when either n_1 or $n_2 > 1$. Figure 4.11(a) shows the initial superquadrupole shape with sharp corners. This shape is made to move with speed $\vec{P} = -\vec{K}$, where \vec{K} is the mean curvature. As shown in figure 4.11(b)–4.11(f), the high curvature regions corresponding to the sharp corners on the surface smoothly diffuse until the superquadrupole collapses into an ellipsoid in figure 4.11(g). Note that an ellipsoid is a superquadrupole with $n_1 = n_2 = 1$. Since the mean curvature on the surface of an ellipsoid is constant, the surface smoothly collapses to a point without changing its shape (see the smaller ellipsoid in figure 4.11(f)). Calculations were done on a $32 \times 32 \times 32$ grid with a time step $\Delta t = 0.001$.

In our last experiment, we remove the shape of a flat superquadrupole using the level set front propagation scheme on 3D image data. In this experiment, results of 32 images each with a particular cross section of the superquadrupole. The image-based speed term \vec{b}_1 is computed from these images according to equation (4.1). A sphere, which is the level surface $\{\phi = 0\}$ of a function $\phi(x, y, z) = x^2 + y^2 + z^2 - 0.04$, forms our initialization (see figure 4.12(a)). This initial surface is moved with speed $\vec{P} = \vec{b}_1$ by updating the value of ϕ as a distance grid according to the details given in algorithm 4. The initial surface expands smoothly in all directions until a portion of it collides with the superquadrupole boundary. At points with high gradients, the \vec{b}_1 values are close to zero and cause the zero set to hardly come to stop near the boundary.

of the superquadrupole shape. This situation is depicted in figures 5 (25),- 5 (27), wherein the initial spherical shape transforms into a flat superquadrupole. Finally, in figure 5 (27), all the points on our shape model are stopped, thereby recovering the initial shape of the flat superquadrupole. Calculations were done on a $32 \times 32 \times 32$ with a time step $\Delta t = 0.0025$.

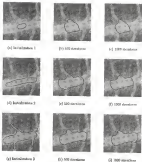


Figure 3.4: An unsuccessful attempt to reconstruct a complex shape with “spoke-and-hub” protrusions using an active contour model. Three different (poor) results are shown in parts (b), (f), & (i) corresponding to three distinct initializations in parts (a), (d) & (g) respectively.



(a) Initial image



(b) After 10 iterations



(c) After 100 iterations



(d) After 200 iterations



(e) After 300 iterations



(f) After 400 iterations

Figure 6.3: Reconstruction of a shape with “significant” protrusions – an artificial test structure. Computations are done on a 50×50 grid with a time step $\Delta t = 0.001$. A narrow band optimum scheme was used with a band width of $\bar{r} = 0.05$.

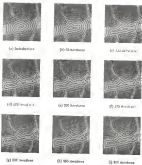


Figure 8.4: Reconstruction of a shape with points. Figure shows the level sets $\{\phi = \text{dist}^2\}_{\phi \in (-R^2 - R_0^2, R_0^2)}$ and $C^2 = 1.05$.



(a) Initial image



(b) After 10 iterations



(c) After 50 iterations



(d) After 100 iterations



(e) After 150 iterations



(f) After 180 iterations

Figure 6.2: Topological split: a single instance of the shape evolution rate flow solution is encountered, the individual shapes. Computation was done on a 64 x 64 grid with a time step $\Delta t = 0.01$.

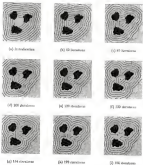


Figure 8.3: Topological split examples: level sets shown are $\{\varphi = \text{dist}^2\}$, $\varphi \in [-8, +8]$ with $C = 0.05$.



(a) Retraction



(b) 1st stroke is on



(c) 2nd stroke is on



(d) Outer boundaries



(e) Second stage



(f) Inner boundaries

Figure 3.3 Shaper with holes, a two-stage scheme is used to arrive at a complete shape description of both simple shapes and shapes with holes. Computation was performed on 128×128 grid and the final step (d) was set to 0.0025.

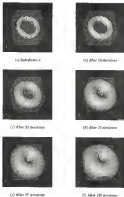


Figure 8.10: Expanding vortex ($\beta = 1$): changing topology is depicted in this example. Calculations were done on a 32×32 grid with a time step $\Delta t = 0.0005$.



(a) Initialization



(b) After 10 iterations



(c) After 100 iterations



(d) After 1000 iterations



(e) After 10000 iterations



(f) After 100000 iterations

Figure 6.11: Motion of a “pushed” exponential disk under its same vibrations. Calculations were done on a $20 \times 30 \times 50$ grid with a time step $\Delta t = 0.001$.



(a) Initial image



(b) After 10 iterations



(c) After 20 iterations



(d) After 30 iterations



(e) After 40 iterations



(f) After 100 iterations

Figure 8.12: Shape recovery in 2D of flat spheroplastic shape. Calculations were done on $m = 120 \times 120 \times 10$ grid with a time step $\Delta t = 0.005$.

CHAPTER 7 CONCLUDING REMARKS

In this dissertation we presented a novel shape modeling scheme. Our approach while retaining the desirable features of existing methods for shape modeling, overcomes most of their drawbacks. We adapt the level-set techniques first introduced in Osher and Sethian [34] to the problem of shape recovery. With this approach, complex shapes can be reconstructed. Unlike the traditional formulations for shape recovery which rely on energy minimization, the final result in our method is not globally independent of the initial state. This is a very desirable feature to have especially in applications such as automatic shape recovery from noisy images. However, our scheme makes no a priori assumptions about the object's topology. Other notable features of our shape modeling scheme include its ability to split and merge freely without any additional bookkeeping during the evolutionary process, and its easy extensibility to higher dimensions. The equations of motion governing our evolutionary system resemble an initial value Hamiltonian Jacobi equation with a gradient right-hand side and are amenable to stable entropy satisfying numerical solution schemes. Thus, the result is a very general shape modeling algorithm which we believe will find numerous applications in the areas of computer vision and computer graphics.

In the context of 2D shape recovery, we have one last bit to add to a step in the rougher kind of object identification by synthesizing a speed function from noisy images. An efficiency is defined to indicate a consistent meaning to this speed term at all points in the computational domain. A significant improvement in time

completely is realized by adopting the narrow-band update strategy. In the narrow-band scheme, the front is advanced by updating the ϕ function in a narrow band around the zero-set. The efficacy and versatility of our method is demonstrated by a series of shape recovery experiments on a set of synthetic and noisy medical images.

Our future research efforts will be focused on extending this method to other applications domains. We have implemented a similar scheme to reconstruct the surface structure of 3D objects from 2D medical image data. However, it is not clear how to adapt the level set formulation in its present form to reconstructed surface shapes from sparse image data. The issue of time complexity is an important one since in 3D, we have seen in chapter 4 that the stability requirements for solving the level set equations forces a very small time-step for fine grids and that the evaluation of the distances is computationally very expensive. The narrow-band strategy has drastically reduced the computational cost in 2D, but implementing a similar scheme in 3D could prove cumbersome. One approach is to embed our level set algorithm in a multiresolution framework, where the loss of accuracy in rapidly-converging coarse grid computations is compensated by highly accurate slowly converging fine grid calculations. Alternatively, sophisticated parallel implementations can be devised. Our work scheme is briefly discussed in chapter 5.

REFERENCES

- [1] W. Aulicrion and J. A. Selsky, "A tensor based approach to joint-3D multi-image for propagating information," *IEEE Report*, Oct. 1993.
- [2] R. Szeliski and E. Rosten, "Multimodality cluster matching," *Computer Vision, Graphics, and Image Processing*, Vol. 48, pp. 1-25, 1989.
- [3] R. Szeliski and P. Szeliski, "Three dimensional object representations revisited," in *Proceedings of First International Conference on Computer Vision*, pp. 331-349, London, England, 1981.
- [4] T. D. Suckling, "Visual Perception by computer," invited talk, *IEEE Systems and Control Conference*, Miami, Florida.
- [5] E. M. Sella and R. C. Vining, "On three-dimensional surface reconstruction methods," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-13, No. 1, pp. 1-14, 1991.
- [6] T. L. Shults and J. R. Fowlkes, "Visual surface reconstruction using sparse depth data," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 65-76, June 1991.
- [7] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, MA.
- [8] H. Blum, "A transformation for reducing non description of shape," *Models for the Perception of Speech and Visual Form* (W. F. Floyd, Ed.), Cambridge MA: MIT Press, 1967.
- [9] L. B. Cohen, "On Active Contour Models and Beltrami," *Computer Vision, Graphics, and Image Processing*, Vol. 53, No. 2, pp. 20-33, March 1991.
- [10] L. B. Cohen and J. Cohen, "Deformable models for 3D rendered images using basic elements and Beltrami," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Urbana, Illinois, June 1992.
- [11] G. Delbortre, and A. Snyck, *Parametric Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1991.
- [12] S. Delgado, M. Robert, and K. Sengul, "Shape representations and image recognition using deformable models," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 467-472, Maui Hawaii, June 1991.
- [13] M. P. De Cossio, *Differential Geometry of Curves and Surfaces*, Prentice Hall Englewood Cliffs, New Jersey, 1977.

- [14] L. C. Eichen and J. Barnick, "Motion of low level sets by mean curvature 1," *Journal of Differential Geometry*, Vol. 52, pp. 423-454, 1991.
- [15] J. A. Brander and A. Murthy, "Clipping based 2D reconstructions of rational free form surfaces: rectangles, ellipses," *IEEE Transactions on Medical Imaging*, Vol. 10 No. 1, pp. 70-80 March 1991.
- [16] W. T. Freeman and J. E. Adelson, "Steepest descent for early vision: shape analysis, and surface decomposition," in *Proceedings of ECCV*, pp. 580-604, Osaka, Japan, 1988.
- [17] M. Kim, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision*, pp. 325-337, 1985.
- [18] B. B. Kagan, A. Terzopoulos, and S. W. Zucker, "Toward a computational theory of shape," *Lecture Notes in Computer Science*, Vol. 123, pp. 582-602, 1984.
- [19] B. B. Kagan, A. R. Terzopoulos, and S. W. Zucker, "Shapes, snakes, and deformations I: The components of shape and motion in feature space," to appear in *International Journal of Computer Vision*, 1991.
- [20] I. Lapinskas and G. Proder, *Numerical Solution of Partial Differential Equations in Science and Engineering*, John Wiley & Sons, New York, 1986.
- [21] D. T. Lee, "Global Area Transformation of a Planar Shape," *IEEE Trans. Pattern Anal. Machine Intel.* PAMI-4, No. 4, pp. 382-393, 1982.
- [22] G. Lee and T. Pockels, "Non-dimensional regularization with denormalization," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, pp. 823-830, 1986.
- [23] F. Leymarie and M. D. Levine, "Imagining the gradient transform using an active contour model," *IEEE Trans. Pattern Anal. Machine Intel.* PAMI-11, No. 5, pp. 56-70, 1989.
- [24] R. Mallik, "Deformable models: Canonical parameters for surface representation and multiple view integration," M.S. Thesis, Department of computer and information sciences, University of Florida, Gainesville, May 1991.
- [25] R. Mallik, J. A. Siskens, and B. C. Yener, "A topology independent shape matching algorithm," in *Proceedings of IEEE Conference on Computer Graphics in Computer Vision II*, San Diego, July 1991.
- [26] R. Mallik, J. A. Siskens, and B. C. Yener, "Shape matching with fixed polygonal area and area approach," Center for Pure and Applied Mathematics, Report PAM 149, Univ. of California, Berkeley August 1991.
- [27] R. Mallik, J. A. Siskens, and B. C. Yener, "A fast level set based algorithm for topology independent shape matching and contouring," UFGV Technical Report TR-90-001, University of Florida, Gainesville, Sept. 1991.
- [28] R. Mallik, J. A. Siskens and B. C. Yener, "Evolutionary trends for topology independent shape matching and contouring," submitted to the *Third European Conference on Computer Vision* Stockholm, Sweden Oct. 1991.

- [20] N. Mays and E. T. Sagan, "Minimal Diagrams of Polygons: a novel and practical algorithm," Tech. Report UP-010-10, 30-33, Univ. of Florida, Gainesville, 1982.
- [21] E. Orlitz and J. A. Sethian, "Finite propagating with numerical gradient speed Algorithms based on Hamilton-Jacobi formalisms," *Journal of Computational Physics*, Vol. 78, pp. 12-48, 1989.
- [22] P. Perona and S.W. Jordan, "Fast robustness, connectivity, consistency and curve detection," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI 11, pp. 422-439, 1989.
- [23] J. P. Phillips, "Fast: Structural description of shape," in *Proceedings of AAAI, Philadelphia, PA*, pp. 495-501, 1980.
- [24] P. Perona and J. Malik, "Scale-space and edge detection using nonmaximal diffusion," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 7, pp. 629-636, 1990.
- [25] W. Press, S. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C*, Cambridge University Press, Cambridge, 1988.
- [26] R. Riesenfeld, "Changes in connectivity in urban random models," *Proceedings of the Workshop on Visual Motion*, pp. 327-343, Irvine, California, March 1989.
- [27] R. Riesenfeld, "Adaptive random control of clumping and ordered perturbations," *Proceedings of ACM Conference on Geometric Methods in Computer Vision*, Vol. 1989, pp. 198-213, San Diego, California, July 1989.
- [28] L.L. Schumaker, "Fitting surfaces to scattered data," in *Approximation Theory II*, G.G. Lorentz, G. B. Chini, and L.L. Schumaker, (eds.) pp. 383-397, Academic Press, New York, 1979.
- [29] J. A. Sethian, "Curvature and the evolution of fronts," *Communs. in Mathematical and Physics*, Vol. 131, pp. 431-458, 1990.
- [30] J. A. Sethian, "Numerical algorithms for propagating interfaces: Hamilton-Jacobi equations and conservation laws," *Journal of Differential Geometry*, Vol. 31, pp. 131-151, 1990.
- [31] R. Strohob and D. Treisman, "Surface matching with minimal particle systems," *Computer Graphics SIGGRAPH*, Vol. 20, No. 3, pp. 185-190, July 1992.
- [32] D. Treisman, "Regularization of inverse visual problems involving discontinuities," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI 8, No. 3, pp. 413-424, 1986.
- [33] D. Treisman, "The computation of visible surface representations," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI 4, Vol. 10, pp. 417-430, 1982.
- [34] D. Treisman, "Matching deformable models to images: Exact and iterative solutions," *Proc. of Optical Society of America Technical Meeting on Machine Vision*, Aspen, Aspen Village, CO, March 1-3, 1987.

- [66] D. Terzopoulos, A. Witkin, and M. Kass, "Elasticity-based modeling and 3D object reconstruction," *Art. Journal of Computer Vision*, Vol. 1, No. 3, pp. 561-573, 1987.
- [67] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on deformable models: Recovering 3D shape and integral motion," *Artificial Intelligence*, 38, pp. 19-73, 1989.
- [68] B. C. Vemuri and E. Mollath, "Deformable models: Optimal parameters for surface representations and multiple view integration," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 728-735, Maui, Hawaii, June 1988.
- [69] B. C. Vemuri and E. Mollath, "Surface fitting with unknown parameters," *Pattern Recognition Letters*, Vol. 12, No. 11, pp. 545-548, November 1992.
- [70] B. C. Vemuri and E. Mollath, "Unknown parameters for surface representations using deformable models," *IEEE Trans. on Systems, Man, & Cybernetics*, Vol. 23, No. 5, pp. 114-122, March/April 1993.
- [71] B. C. Vemuri and E. Mollath, "Constructing optimal parameters with active models for learning surface representations," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 7, pp. 684-694, July 1993.
- [72] B. C. Vemuri, A. Muelick, and J. E. Aggarwal, "Curvature-based representation of objects from range data," *Art. Journal of Image and Vision Computing*, 6, pp. 107-114, 1988.
- [73] Y. F. Wang and J. F. Wang, "Surface reconstruction using deformable models with interior and boundary constraints," in *Proceedings of ACCV*, pp. 307-322, Osaka, Japan, 1990.
- [74] D. N. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.

BIOGRAPHICAL SKETCH

Purushotham Mallik was born at Vaypattam, India, in 1933. He received his B-Eng degree with honors in electrical engineering and M.Sc. degree in physics from the Institute of Technology and Science, India, in 1955. In summer of 1955 he worked at the National Geophysical Research Institute, India. During spring of 1956 he held a research assistant position with the semiconductor physics group in Central Electrical and Electronics Research Institute, India, where he was involved in modeling the response and development of positive photoconductor. He subsequently received the M.S. degree in 1957, and expects to receive the Ph.D. degree in computer and information sciences from the University of Florida, Gainesville, in 1963. Since the summer of 1959 he was a graduate research assistant at the University of Florida, and was affiliated to the Center for Computer Vision and Visualization. His research interests are focused on computational vision, shape modeling, surface reconstruction, computer graphics, image processing, and medical image interpretation.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy


Bill C. Barnes, Chairman
Associate Professor of Computer and
Information Science

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy


James R. Bellamy, Chairman
Professor of Mathematics
University of California at Berkeley

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy


Edward R. Bittner
Professor of Computer and
Information Science

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy


Andrew F. Davis
Assistant Professor of Computer and
Information Science

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Paul A. Fodorak
Associate Professor of Computer and
Information Systems

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Sarah Eas
Professor of Mathematics

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December 1988



Richard M. Phillips
Dean, College of Engineering

James A. Hollerbach
Dean, Graduate School